ARMY RESEARCH LABORATORY

Modeling Nonmonotonic Dose-Response Curves

Quinton J. Nottingham, Jeffrey B. Birch, and Barry A. Bodt

ARL-TR-1972 January 2001

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5425

ARL-TR-1972 January 2001

Modeling Nonmonotonic Dose-Response Curves

Quinton J. Nottingham, Jeffrey B. Birch
Virgina Polytechnic Institute & State University

Barry A. Bodt
Information Science and Technology Directorate, ARL

Abstract

A number of procedures have been used to analyze nonmonotonic binary data to predict the probability of response. Some classical procedures are the Up and Down strategy, the Robbins-Monro procedure, and other sequential optimization designs. Recently, nonparametric procedures such as kernel regression and local linear regression have been applied to this type of data. It is well known that kernel regression has problems fitting the data near the boundaries, and a drawback with local linear regression is that it may be too linear when fitting data from a curvilinear function. This report introduces a procedure called local logistic regression, which fits a logistic regression function at each of the data points. United States Army projectile data are used in an example that supports the use of local logistic regression for analyzing nonmonotonic binary data for certain response curves. Properties of local logistic regression are presented along with simulation results that indicate some of the strengths of the procedure.

Contents

1 Introduction 1

2 Possible Models 3

2.1 Parametric Models 3

2.2 Nonparametric Models 5

3 Some Simulation Results 11

4 Discussion and Summary 13

Bibliography 14

Appendices 17

A Numerical Considerations 17

B Program Documentation & Comments 19

Distribution 33

Report Documentation Page 35

Figures

1 Army macrocomposite data, illustrating shatter gap phenomenon 2

2 Linear logistic regression fit and local linear regression fit to macrocomposite data 4

3 Comparison of fits between llogrs and cdf mixture model 5

4 Local logistic regression fit to macrocomposite data with 95% confidence bands 9

5 Local logistic regression fit to macrocomposite data using linear predictor for confidence interval 9

6 Curves 1 to 3 from table 1 generated for Monte Carlo simulation 11

Tables

1 Parameter values for simulated curves 11
2 Lower V_{50} estimation summary 12
3 Upper V_{50} estimation summary 12

1. Introduction

Consider modeling the dose-response curve for the ungrouped binary response variable y. This dose-response curve represents the probability that y is 1 for each given value of " dose," the single regressor v. In many applications in such areas as biology, industry, and business, this response curve may be modeled successfully by a monotonic parametric function such as the normal or logistic cumulative distribution function (cdf). However, in this report, we present an application where the response curve is nonmonotonic, rendering traditional methods such as logistic regression inappropriate. We suggest the use of local logistic regression (llogr), a nonparametric method that simultaneously enables modeling a nonmonotonic dose-response curve while maintaining the restriction that all estimated responses take values between 0 and 1. The following real-life example demonstrates the flexibility and applicability of this new procedure.

An experiment of importance to the development of a kinetic energy penetrator (a " projectile") and the armor to resist it is to model the probability that the projectile will penetrate the plate of armor as a function of the velocity of the projectile. Normally, one expects the probability of penetration to increase as velocity increases. Routinely, testers are asked to estimate the V_{50}, the velocity where the probability is 0.50 that the projectile will penetrate the armor. Experimenters use variants of the Up and Down strategy (Dixon and Mood, 1948), the Robbins-Monro (Robbins and Monro, 1951), or other sequential designs to gather observations, and the maximum likelihood estimate (MLE) for the mean (V_{50}) is formed with a logistic or normal response function. An interlaboratory study of V_{50} estimation revealed widely varying estimates for some penetrator-armor matches (Chang and Bodt, 1997). Initially, the blame was thought to lie with varying test procedures among experimental testing sites. Recent careful study of the data suggests a phenomenon at work here that one material scientist refers to as the " shatter gap."

The shatter gap is not precisely defined, and the exact physical mechanism for significant bullet shattering is unknown. Simply put, when bullet shattering occurs against certain armor materials, kinetic energy is diffused, thereby reducing the ability of the penetrator to defeat the armor. This reduction is reflected over a range of velocity in which a decrease in response probability accompanies increasing velocity. Eventually, as velocity continues to increase, the kinetic energy developed is so much an overmatch that the probability of penetration rises again. One working definition, suitable for data currently available, characterizes the shatter gap in terms of an upper and lower V_{50}, V_{50U}, and V_{50L}, respectively. In those data, penetrator shatter does not cause the probability of penetration to decrease until after

a V_{50} is achieved. A second, higher, V_{50} is achieved by partially intact penetrators with increased velocity. The interval they bound is thought of as the shatter gap.

These ideas are illustrated in figure 1, showing the results of firing 303 projectiles at macrocomposite armor with velocities ranging from 1432 to 2773 ft/s. The response variable is whether the projectile penetrated the armor (scored as a "1") or failed to penetrate the armor (scored as a "0"). Typical of penetrator-against-plate response curves, low velocities result in a low proportion of penetrations, and high velocities result in a high proportion of penetrations. Unusual is that over intermediate velocities, the proportion of penetrations first rises with velocity, then falls, and then rises once more. Note especially the response activity between 2100 and 2300 ft/s relative to response activity for neighboring velocities.

The shatter gap phenomenon is more clearly seen from the curve in figure 1; this curve (obtained by the method presented in sect. 2) represents the estimated probability of penetration as a function of velocity.

We can represent the V_{50} graphically by extending a line at the probability of 0.5 over to the estimated probability curve and dropping lines down to the velocity axis at the intersection of the line and the curve. The interval bounded by these two V_{50}'s on the velocity axis is the shatter gap. Figure 1 shows that the shatter gap extends from V_{50L} of approximately 1713 ft/s to V_{50U} of approximately 2510 ft/s, a distance of 797 ft/s, far larger than the usual error of approximately 100 ft/s.

In section 2 we discuss several analytical methods to deal with data of this type.

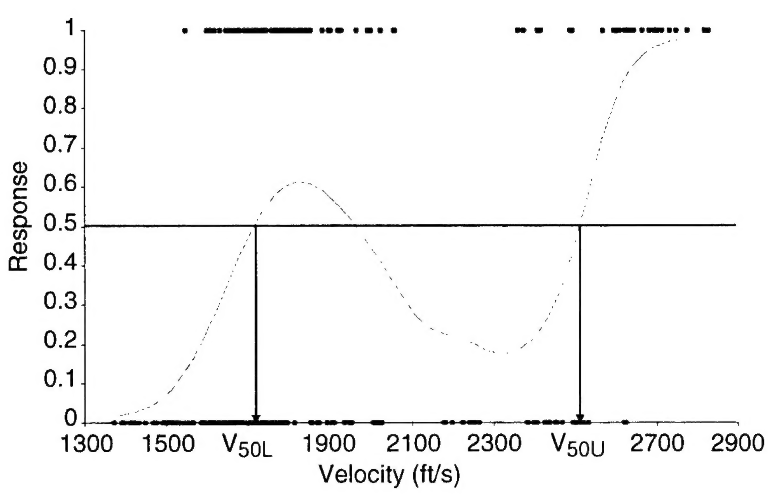


Figure 1. Army macrocomposite data (solid dots), illustrating shatter gap phenomenon. Curve is estimated probability of penetration.

2. Possible Models

An obvious approach to fitting 0 to 1 response data is to use the standard logistic regression model. However, as will be demonstrated, such a model will fail to identify the shatter gap in the penetration data because of the inherent monotonically increasing (or decreasing) nature of the logistic curve.

Two more promising possible models for fitting data of the type given in the curve in figure 1 are the cdf mixture model (CDFMM) (Bodt and Chang, 1997), a parametric method model procedure, and local llogr, a nonparametric method.

2.1 Parametric Models

Consider the model

$$y_i = P(v_i) + \varepsilon_i, \quad i = 1, ..., n \ ,$$

where y_i represents the response variable, $P(v_i)$ is the penetration probability that is a function of the velocity v_i at the ith experimental run, and ε_i is the random error term. The response is recorded as a " 1" if armor penetration has occurred and a " 0" otherwise. The errors are assumed to have expectation of zero.

In the logistic regression model, the form of $P(v_i)$ is based on the logistic cumulative distribution function, where

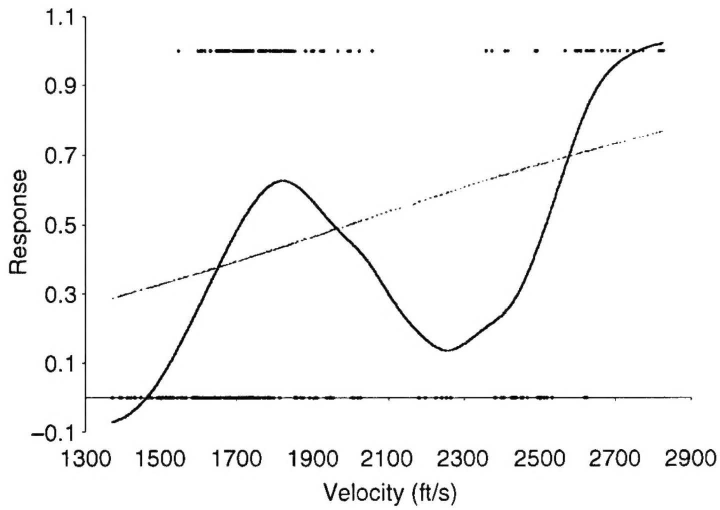
$$P(v_i) = \left(1 + \exp\left[-\left(\beta_0 + \beta_1 v_i\right)\right]\right)^{-1} = \left(1 + \exp\left(-\underline{v}_i'\underline{\beta}\right)\right)^{-1} = F\left(\underline{v}_i'\underline{\beta}\right),$$

where $\underline{v}_i' = (1 \quad v_i)$ and $\underline{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$, where underlining indicates a vector and a prime indicates a transpose.

The method of maximum likelihood is used to obtain estimates (MLEs), of the unknown parameters β_0 and β_1. To obtain the MLEs, the method requires an iterative procedure, for example, the Gauss-Newton method using iterated reweighted least squares (IRLSs). Details concerning logistic regression may be found in McCullagh and Nelder (1983), Myers (1990) and Ryan (1997), among many others.

While the logistic regression fit guarantees an estimated response between 0 and 1, the fit, as seen in figure 2 (dashed line), is entirely inadequate to the armor penetration data- resulting in a nearly flat line fit, completely missing the up and down nature of the response seen in the llogr fit (solid line).

Figure 2. Linear logistic regression fit (dashed line) and local linear regression fit (bold solid line) to macrocomposite data.



The CDFMM approach, using mixtures of three cdfs, models the penetration probability $P(v)$ as a function of velocity v as

$$P(v) = [1 - F_T(v; \mu_T, \sigma_T)]F_I(v; \mu_I, \sigma_I) + F_T(v; \mu_T, \sigma_T)F_S(v; \mu_S, \sigma_S)$$

where F_T, F_I, and F_S represent appropriately chosen cdfs for the transition, intact, and shattered mechanisms.

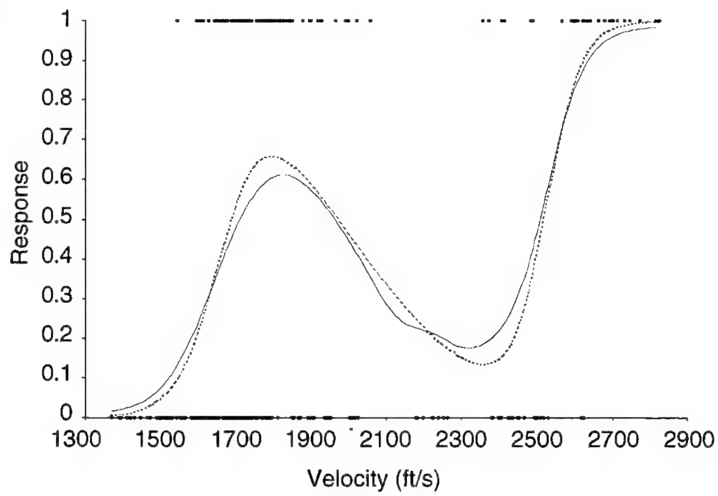
One interpretation of the above model is to consider $P(v)$ as the probability of response due to the mixture of two penetration mechanisms. The first mechanism is penetration from an intact (I) projectile. The second mechanism is penetration from a shattered (S) projectile. The penetrator being intact or shattered is also a function of velocity in this formulation, and so the proportions of the two penetration mechanisms in the mixture will vary also with velocity. It is assumed in the model that the probability of projectile shatter will increase monotonically. This proportion change is modeled as the transition (T) from the intact mechanism to the shattered mechanism.

The CDFMM model fit to the Army penetration data is illustrated in figure 3, which also shows for comparison the llogr fit.

The dashed line represents the CDFMM fit using logistic cdfs with estimated means of 1972, 1664, and 2523 and estimated standard deviations of 186, 56, and 46 for the transition, intact, and shattered mechanisms, respectively. The curves are obviously quite similar, with the CDFMM fit resulting in slightly larger (at around 1800 ft/s) and slightly smaller (at around 2400 ft/s) penetration probabilities. The shatter gap obtained with the CDFMM fit is only slightly wider than that obtained with the llogr fit. The estimates are the MLEs for this nonlinear model with binary response, an example of a generalized nonlinear model.

One problem with the CDFMM approach is that six parameter estimates are required to fit the model. This requires considerable effort to achieve convergence. We used the SAS Proc NLIN (SAS Version 7) with an appropriately iterated weight matrix to obtain the fit for the Army penetration

Figure 3. Comparison of fits between llogrs (solid line) and cdf mixture model (dashed line).



data. Our experience suggests that this procedure is extremely sensitive to the starting values, often failing to converge if the starting values are only slightly different from the values at convergence. Additionally, the resulting fit, as can be seen in figure 3, is somewhat inflexible, resulting in fits that look like a mixture of logistic cdfs, even when the data do not exhibit such a pattern. That is, the curve may appear to be too smooth, as illustrated in figure 3, where the llogr method seems to capture the wiggle resulting from the cluster of data around 2250 ft/s.

2.2 Nonparametric Models

Another approach to modeling these data is to use nonparametric regression techniques. Such methods as kernel regression and local linear regression can trace out smooth curves with a multitude of peaks and valleys. The shatter gap effect that must be captured in the armor penetration example could then be identified by such " local" fi tting techniques.

Consider again the model

$$y_i = P(v_i) + \varepsilon_i, \quad i = 1, ..., n \ ,$$

where $P(v_i)$ is no longer specified by a parametric function such as the CDFMM function given in section 2.1. It is assumed only that $P(v_i)$ is some arbitrary smooth function. If $E(\varepsilon_i) = 0$, then $E(y_i) = P(v_i) = P_i$, the armor penetration probability at velocity v_i. It follows that $\hat{y}_i = \hat{P}(v_i) = \hat{P}_i$ is the estimated penetration probability at velocity v_i. Kernel regression is a method of approximating $P(v_0)$ (where v_0 is any arbitrary velocity including v_i, the velocity for the ith observation) using a weighting sequence on the response variable, where the weights are functions of the distances between the values of the regressor variable and v_0. One form of the kernel weighting sequence, proposed by Nadaraya (1964) and Watson (1964), assigns weights of h_{0j}^k to y_j when estimating the response at v_0 by

$$h_{0j}^k = \frac{K\left(b^{-1}\left(v_0 - v_j\right)\right)}{\sum\limits_{i=1}^{n} K\left(b^{-1}\left(v_0 - v_j\right)\right)} \ ,$$

where $K()$ represents some appropriate kernel function and b is bandwidth. The kernel estimate of response at v_i, $\hat{P}_i^k$, is given by a weighted average of the observed y_i's as

$$\hat{P}_i^k = \sum_{j=1}^{d} h_{ij}^k y_j \ .$$

where h_{ij}^k (v_i replaces v_0 in the h_{0j}^k expression above) are the kernel weights on the observations y_j for estimating the mean response at velocity v_i using the kernel function $K()$, $j = 1, 2, ..., n$.

All n estimates of mean response may be expressed as $\underline{\hat{y}}^k = H^k \underline{y}$, where $\underline{\hat{y}}^k$ is the $n \times 1$ vector of kernel predictions and H^k is the $n \times n$ matrix of weights with rows $\underline{h}_i^{k\prime} = \left(h_{i1}^k \quad h_{i2}^k \quad \ldots \quad h_{in}^k\right)$, for $i = 1, ..., n$. The matrix H^k, called the kernel hat matrix, plays a role in kernel regression similar to that of the ordinary least square (OLS) "hat" matrix H in linear regression.

The kernel estimate of $P(v_i)$ depends on the choice of the kernel function $K()$ and the bandwidth b. For example, one popular choice for kernel function is the simple Gaussian kernel defined as $K(u) = \exp\left(-u^2\right)$. The method for selecting the bandwidth is extremely critical. The magnitude of the bandwidth determines the smoothness, or lack thereof, of the regression function. One of the more commonly used methods, referred to as the method of cross-validation, selects the bandwidth to minimize the PRESS (Allen, 1974) statistic. A related method finds b by minimizing a penalized PRESS statistic, called PRESS*(b), given by

$$\text{PRESS}^*(b) = \frac{\sum_{i=1}^{n} \left(y_i - \hat{P}_{i,-i}^k(b)\right)^2}{n - \text{tr}\left[H^k\right]} \ ,$$

where $\hat{P}_{i,-i}^k(b)$ is the "minus-i" predicted penetration probability based on kernel regression v_i for the current value of b with the ith observation removed, and $\text{tr}\left[H^k\right]$ is the trace of the $n \times n$ kernel weight matrix H^k. Since $\text{tr}\left[H^k\right]$ reflects the kernel fits' "model degrees of freedom" (Cleveland, 1978), it is seen that the denominator of PRESS*(b) penalizes the PRESS statistic for choosing b too small. Empirical studies by the authors and others (Einsporn and Birch, 1993; Mays, 1995) have demonstrated that using PRESS*(b) is often superior to using PRESS as a bandwidth selector.

The kernel regression fit to the armor penetration data results in a smooth curve that follows the up and down pattern exhibited in figures 1 and 3. While this is a desirable characteristic, the resulting curve is not appealing, since it is entirely possible that some of the curve's fitted values may lie outside the 0 to 1 range, a natural restriction resulting from estimating the penetration probabilities. Indeed, the kernel regression fit to the Army macrocomposite data results in fits at the lower left that are less than zero and fits to the upper right that exceed one. This problem is not avoided by the use of more complicated nonparametric smoothers, such as local linear regression (Hardle, 1990), where each fit at $v = v_0$ is obtained locally by a weighted

simple linear regression model with the h_{0j}^k's serving as the weights, or local polynomial regression (Fan and Gijbels, 1996). These methods also do not restrict the resulting fits to between 0 and 1, a requirement when modeling probabilities. The local linear regression fit to the macrocomposite data can be seen in figure 2.

However, the logistic regression parametric method can be combined with the nonparametric concept of local fitting to produce a smooth curve, flexible enough to capture the up and down patterns exhibited in the armor penetration data and giving fits that are between zero and one. We propose local (linear) logistic regression where at each velocity $v = v_0$, a weighted linear logistic fit is obtained with h_{0j}^k's serving at the weights, in exactly the same manner as in local linear regression. That is, the fit at $v = v_0$ would be obtained as

$$\hat{P}(v_0) = \left(1 + \exp\left[-\left(\hat{\beta}_{00} + \hat{\beta}_{10} v_0\right)\right]\right)^{-1} = \left(1 + \exp\left[-\left(\underline{v}_0' \hat{\underline{\beta}}_0\right)\right]\right)^{-1}.$$

The estimated coefficients $\hat{\underline{\beta}}_0$, which change values for each $v = v_0$, are obtained via the same IRLS algorithm referred to earlier for logistic regression, with a slight adjustment to the weight matrix. Thus, one step of the algorithm would compute the updated value of $\hat{\underline{\beta}}_0$ as

$$\hat{\underline{\beta}}_{01} = \hat{\underline{\beta}}_{00} + \left(X' \hat{W}_0 X\right)^{-1} X' \hat{W}_0 \underline{y}_0^*,$$

where the $n \times n$ diagonal matrix $\hat{W}_0$ has elements

$$\hat{w}_{jj} = F\left(\underline{v}_j' \hat{\underline{\beta}}_{00}\right)\left(1 - F\left(\underline{v}_j' \hat{\underline{\beta}}_{00}\right)\right) h_{0j}^k, \quad \text{for } j = 1, \ldots, n.$$

We see that llogr follows a weighting scheme that combines the logistic weights with the kernel weights in a local manner. The $n \times 1$ vector $\underline{y}_0^*$ has elements

$$y_{0j}^* = \frac{y_j - F\left(\underline{v}_j' \hat{\underline{\beta}}_{00}\right)}{f\left(\underline{v}_j' \hat{\underline{\beta}}_{00}\right)},$$

the locally adjusted response to velocity v_j. Upon convergence of the IRLS algorithm, $\hat{\underline{\beta}}_0$ is set equal to $\hat{\underline{\beta}}_{01}$, and the estimated response is obtained as $\hat{P}(v_0) = F\left(\underline{v}_0' \hat{\underline{\beta}}_0\right)$.

Approximate inferential information can be obtained by straightforward application of the variance operator, the multivariate delta method, and the asymptotic distribution of the MLEs. For example, it can be shown that the asymptotic variance of $\hat{\underline{\beta}}_0$ is $\text{var}\left(\hat{\underline{\beta}}_0\right) = (X'W_0X)^{-1}(X'W_1X)(X'W_0X)^{-1}$, where the diagonal elements of the $n \times n$ diagonal matrix W_0 are $w_{0_{jj}} = F\left(\underline{v}_j' \underline{\beta}_0\right)\left(1 - F\left(\underline{v}_j' \underline{\beta}_0\right)\right) h_{0j}^k$, and the diagonal elements of the $n \times n$ diagonal matrix W_1 are $w_{1_{jj}} = w_{0_{jj}} h_{0j}^k$. Using this result, one can express the asymptotic variance of each fit at $v = v_0$ by applying the delta method as

$$\text{var}\left(F\left(\underline{v}_0' \hat{\underline{\beta}}_0\right)\right) = f^2\left(\underline{v}_0' \underline{\beta}_0\right) \underline{v}_0' \,\text{var}\left(\hat{\underline{\beta}}_0\right) \underline{v}_0.$$

It follows then from the asymptotic distribution of $F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right)$ that one form of the approximate $(1-\alpha)\,100\%$ confidence interval for $P\,(v_0)$ is

$$F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right) \pm z_{1-\frac{\alpha}{2}}\ \mathrm{se}\left(F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right)\right),$$

where the standard error of $F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right)$ is the square root of $\mathrm{var}\left(F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right)\right)$, with $\hat{\underline{\beta}}_0$ replacing $\underline{\beta}_0$ in this variance expression.

While this approach for establishing confidence intervals is a straightforward application of Wald inference, the resulting interval, (LCL, UCL), may result in lower confidence limit (LCL) less than zero or an upper confidence limit (UCL) greater than one. To avoid this problem, we can consider a second approach where we obtain the $(1-\alpha)\,100\%$ confidence interval for the linear predictor $\underline{v}_0'\hat{\underline{\beta}}_0$ by once again applying the delta method. It can be shown that the asymptotic variance of the linear predictor $\underline{v}_0'\hat{\underline{\beta}}_0$ is

$$\mathrm{var}\left(\underline{v}_0'\hat{\underline{\beta}}_0\right) = \underline{v}_0'\ \mathrm{var}\left(\hat{\underline{\beta}}_0\right)\underline{v}_0$$

with resulting $(1-\alpha)\,100\%$ confidence interval as

$$\underline{v}_0'\hat{\underline{\beta}}_0 \pm z_{1-\frac{\alpha}{2}}\ \mathrm{se}\left(\underline{v}_0'\hat{\underline{\beta}}_0\right) = (\mathrm{LCL}^*, \mathrm{UCL}^*).$$

It follows that an approximate $(1-\alpha)\,100\%$ confidence interval for $F\left(\underline{v}_0'\hat{\underline{\beta}}_0\right)$ is $(F\,(\mathrm{LCL}^*),\ \ F\,(\mathrm{UCL}^*))$.

The general procedure is to obtain the fit and corresponding $(1-\alpha)\,100\%$ confidence interval (by either method) for a fine grid of values of v_0 throughout the range of velocity values. Connecting the fits, lower confidence limits, and upper confidence limits with straight line segments results in three smooth curves, the curve fit to the data and the two $(1-\alpha)\,100\%$ confidence bands. Both types of interval methods are illustrated in figures 4 and 5. The method based on the linear predictor results in unsatisfactory wide intervals at the boundaries.

Our procedure is a specific example, though developed independently, of the related method, local polynomial kernel regression of generalized linear models, introduced by Fan, Heckman, and Wand (1995).

Once the response curve and the confidence bands are obtained, the V_{50} as well as confidence intervals for the V_{50} can be computed. Because it is not possible to obtain a closed functional form for the estimate of $P\,(v_0)$, the V_{50} and confidence intervals for the V_{50} are obtained through the "inverse" process, where a horizontal line is extended from $P(v) = 0.5$ to the three curves, representing the upper confidence limit, the estimated response curve, and the lower confidence limit. At the intersection, vertical lines are dropped to the velocity axis, where the V_{50} and the 95-percent confidence interval for the V_{50}, (V_{50L}, V_{50U}), can be obtained. (We note that if the shatter gap effect is present, there will be two V_{50} values, each with

Figure 4. Local logistic regression fit to macrocomposite data with 95% confidence bands.

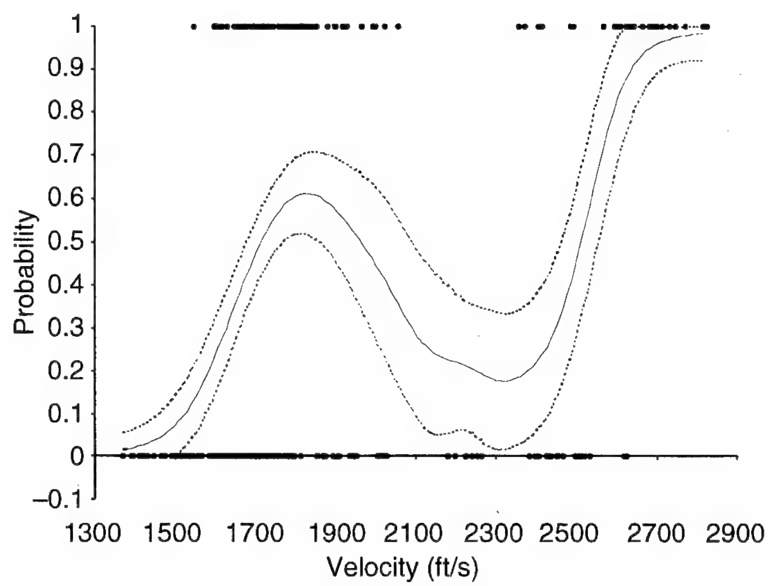
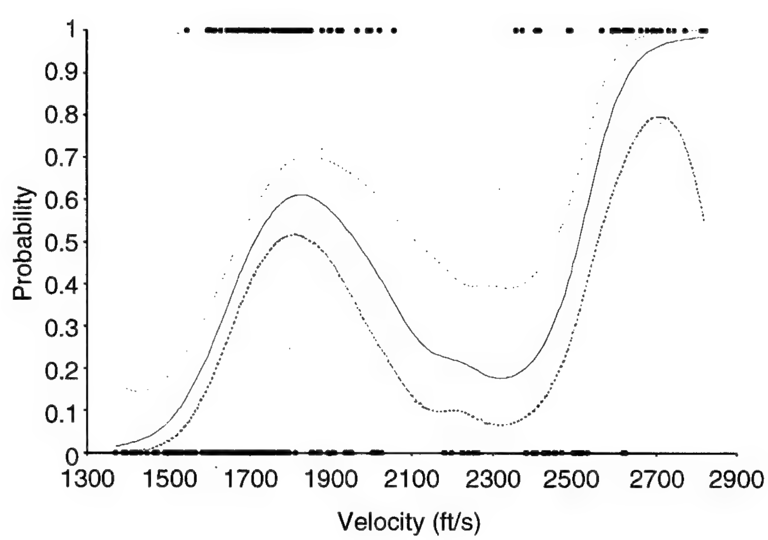


Figure 5. Local logistic regression fit to macrocomposite data using linear predictor for confidence interval.



a corresponding 95-percent confidence interval.) This process is easily accomplished numerically with a computer. The IRLS algorithm to compute the llogr fit is outlined in appendix A. A SAS macro for calculating the llogr fit to any set of 0 to 1 data may be obtained from the authors. The complete SAS macro is listed in appendix B.

3. Some Simulation Results

In this section, we present a small simulation study, generating binary data from several nonmonotonic dose-response curves. Because the primary interest with the Army penetration data is estimation of the V_{50} values, we evaluate the llogr procedure by comparing the estimated V_{50} values with the true V_{50} values. The true cdf used in this evaluation was chosen as the cdf mixture model:

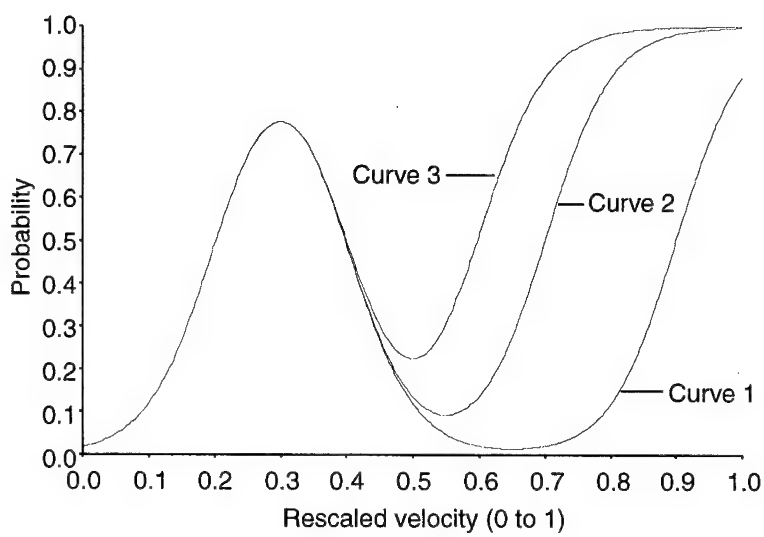
$$P(v) = [1 - F_T(v; \mu_T, \sigma_T)] F_I(v; \mu_I, \sigma_I) + F_T(v; \mu_T, \sigma T) F_S(v; \mu_S, \sigma_S).$$

The values of μ_T, μ_I, and μ_S were varied while σ_T, σ_I, and σ_S were set equal to each other at a constant value of 0.05. The cdf representing F_T, F_I, and F_S was chosen to be the logistic cdf. For example, the cdf representing F_T can be written as $F_T = (1 + \exp(-(x - \mu_T)/\sigma_T))^{-1}$. While our full simulation study considered a variety of values of μ_T, μ_I, and μ_S, for the sake of brevity we show only the results for the three combinations given in table 1 and used to generate the three curves displayed in figure 6. As seen in figure 6, each curve has two V_{50} values.

Table 1. Parameter values for simulated curves.

Curve	μ_T	μ_I	μ_S
1	0.4	0.2	0.9
2	0.4	0.2	0.7
3	0.4	0.2	0.6

Figure 6. Curves 1 to 3 from table 1 generated for Monte Carlo simulation.



For the simulations, random binary responses were generated for 100 evenly spaced velocities between the values of 0 and 1. As can be seen in figure 6, the velocity values have been rescaled between 0 and 1 so that the value of the bandwidth would be more meaningful. The cdf mixture model was simulated with SAS interactive matrix language (IML) for each of the curves in table 1. The process was repeated 500 times, and the average V_{50} values were computed over the 500 Monte Carlo repetitions. We view these average V_{50} values as assessing the ability of the llogr procedure to estimate the true V_{50} values. We used the PRESS* procedure here to obtain the bandwidth for the llogr procedure.

Tables 2 and 3 contain the results from the simulations for estimating the V_{50} values. As can be seen in the fifth column of the tables, the local logistic regression does an exceptional job in estimating the lower and upper V_{50} values when they are calculable. The average absolute error of each estimate is less than 0.03. The quality of these estimates indicates the goodness of fit of the llogr procedure and its ability to estimate the V_{50} values. The excellent estimation can also be attributed to the ability of the PRESS* procedure in finding an appropriate value of the bandwidth to fit this family of curves.

Note also that in the third column of tables 2 and 3, the number of V_{50} values that were not calculable is in parentheses. In order for both V_{50L} and V_{50U} to be estimated, the fitted llogr curve must cross the $P(v) = 0.5$ line three times, as illustrated in figure 1. When the llogr curve crosses the $P(v) = 0.5$ curve line only once, then only one of the V_{50} values can be estimated: V_{50L} if the crossing occurs at lower values of velocity, and V_{50U} if the crossing occurs at upper values of velocity (with the other V_{50} value being incalculable). Data generated from a curve similar to curve 1 in figure 6 will more frequently fail to result in a both V_{50L} and V_{50U} being estimable. Low penetration probabilities for high velocities in curve inhibit the movement of the sequential design to the highest velocities where a third crossing of $P(v) = 0.5$ could occur. Note that 31 percent of the time (156 samples out of 500), the V_{50U} could not be calculated for curve 1: clearly an undesirable situation. One way to resolve this problem would be to increase the sample size at larger velocities.

Table 2. Lower V_{50} estimation summary (incalculable values in parentheses).

Curve	True V_{50L}	Estimate V_{50L}	Average error	Average abs. error
1	0.20190	0.20827 (41)	0.00637	0.02590
2	0.20190	0.20782 (38)	0.00592	0.02425
3	0.20190	0.20901 (45)	0.00711	0.02363

Table 3. Upper V_{50} estimation summary (incalculable values in parentheses).

Curve	True V_{50L}	Estimate V_{50L}	Average error	Avgerage abs. error
1	0.89999	0.88810 (156)	−0.01190	0.02504
2	0.69975	0.69845 (0)	−0.00130	0.01834
3	0.59810	0.58604 (22)	−0.01205	0.02672

4. Discussion and Summary

The research presented here has shown that the llogr procedure provides a more flexible and reasonable approach to fitting binary data from a non-monotonic function than the classic logistic regression procedure, cdf mixture models, and some other nonparametric procedures (kernel and local linear regression). Through simulating a family of nonmonotone functions, it has been shown that the llogr procedure provides an outstanding estimate of V_{50} values with very small average and absolute errors, when calculable.

Bibliography

Allen, D. A. (1974). The relationship between variable selection and data augmentation and a method of prediction. *Technometrics* **16**, 125–127.

Chang, A. L., and Bodt, B. A. (1997). *JTCG/AS Interlaboratory Ballistic Test Program-Final Report.* U.S. Army Research Laboratory, ARL-TR-577.

Chu, C.-K., and Marron, J. S. (1991). Choosing a kernel regression estimator. *Stat. Sci.* **6**, 4, 404–436.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.* **74**, 368, 829–836.

Copas, J. B. (1983). Plotting p against x. *Appl. Stat.* **32**, 1, 25–31.

Dixon, W. J,. and Mood, H. M. (1948). A method for obtaining and analyzing sensitivity data. *J. Amer. Stat. Assoc.* **23**, 109–126.

Einsporn, R. L., and Birch, J. B. (1993). *Model-Robust Regression: Using Nonparametric Regression to Improve Parametric Regression Analyses.* Virginia Polytechnic Institute and State University, Technical Report 93-5.

Fan, J., Heckman, N. E., and Wand, M. P. (1995). Local polynomial kernel regression for generalized linear models and quasi-likelihood functions. *J. Amer. Stat. Assoc.* **90**, 429, 141–150.

Fan, J., and Gijbels, I. (1992). Variable bandwidth and local linear regression smoothers. *Ann. Stat.* **20**, 2008–2036.

Härdle, W. (1990). *Applied Nonparametric Regression.* Cambridge University Press, New York.

Mays, James (1995). *Model-Robust Regression Combining Parametric, Nonparametric, and Semiparametric Methods.* Unpublished Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

McCullagh, P., and Nelder, J. A. (1983). *Generalized Linear Models.* New York: Chapman and Hall.

Myers, R. H. (1990). *Classical and Modern Regression with Applications* (2nd edition). PWS-Kent, Boston, Massachusetts.

Nadaraya, E. A. (1964). On estimating regression. *Theory Probab. Its Appl. USSR* **9**, 141–142.

Nottingham, Q. J., Birch, J. B., and Bodt, B. A. (2000). Local logistic regression: An application to Army penetration data, *J. Stat. Comput. Simu.* **66**, 35–50.

Robbins, H., and Monro, S. (1951). A stochastic approximation method. *Annals Math. Stat.* **22**, 400- 407.

Ryan, T. P. (1997). *Modern Regression Methods.* New York: John Wiley and Sons.

Wahba, G., and Wold, S. (1975). A completely automatic French curve: fitting spline functions by cross-validation. *Commun. Stat. Ser. A* **4**, 1- 17.

Appendix A. Numerical Considerations

The values of the fits are obtained by using the IRLS algorithm. The procedure is outlined below.

To find the proper bandwidth:

For each fixed value of bandwidth b, perform the following:

1. Compute the kernel weights for each value of velocity, $v = v_i$, for $i = 1, \ldots, n$, as

$$h_{ij}^k = \frac{K\left(\frac{v_i - v_j}{b}\right)}{\sum_{j=1}^{n} K\left(\frac{v_i - v_j}{b}\right)}.$$

2. At each $v = v_i, i = 1, \ldots, n$, compute initial coefficients via local linear regression (llr) as

$$\underline{\hat{\beta}}_{i0} = \left(X'\hat{W}_i X\right)^{-1} X'\hat{W}_i \underline{y}\,,$$

where the $n \times n$ diagonal matrix $\hat{W}_i$ has elements $\hat{w}_{jj} = h_{ij}^k$.

3. Compute $n \times 1$ vector $\underline{y}_i^*$ as $y_{ij}^* = \frac{y_j - F\left(\underline{v}_j'\underline{\hat{\beta}}_{i0}\right)}{f\left(\underline{v}_j'\underline{\hat{\beta}}_{i0}\right)}$ and update the elements of $\hat{W}_i$ as

$$\hat{w}_{jj} = F\left(\underline{v}_i'\underline{\hat{\beta}}_{i0}\right)\left(1 - F\left(\underline{v}_i'\underline{\hat{\beta}}_{i0}\right)\right) h_{ij}^k, \quad \text{for } j = 1, \ldots, n.$$

4. Compute the updated estimates of the coefficients as

$$\underline{\hat{\beta}}_{i1} = \underline{\hat{\beta}}_{i0} + \left(X'\hat{W}_i X\right)^{-1} X'\hat{W}_i \underline{y}_i^*.$$

5. If $\underline{\hat{\beta}}_{1i} \approx \underline{\hat{\beta}}_{0i}$, quit and let $\underline{\hat{\beta}}_i = \underline{\hat{\beta}}_{1i}$. Otherwise, replace $\underline{\hat{\beta}}_{0i}$ by $\underline{\hat{\beta}}_{1i}$ and return to step 3 above.

6. After convergence, compute the fit at $v = v_i$ as $\hat{P}(v_i) = F\left(\underline{v}_i'\underline{\hat{\beta}}_i\right)$.

7. After the fits are obtained for all n observations, compute the PRESS*(b) statistic as

$$\text{PRESS}^*(b) = \frac{\sum_{i=1}^{n}\left(y_i - \hat{P}(v_i)_{-i}\right)^2}{n - \operatorname{tr}\left[H^k\right]},$$

where $\hat{P}(v_i)_{-i}$ is the llogr (local logistic regression) fit based on the current bandwidth b, obtained with the ith data point (y_i, v_i) removed from the data set. The current version of the algorithm obtains $\hat{P}(v_i)_{-i}$ by using steps 1 through 6 above, based on the $n - 1$ point data set.

8. Determine b^*, the value of b that minimizes PRESS*(b). This can be accomplished either through a search routine (the algorithm currently uses a binary search to find b^*) or by finding b^* through a candidate list of bandwidths.

Upon completion of steps 1 though 8, the value of b^* has been found and the n fits are obtained at the n data points based on b^*. Often, however, the fitted curve is desired over a second series of v values, which may or may not include the original data points. For example, the predicted probability of response may be desired for all velocities ranging from the minimum observed velocity to the maximum observed velocity at every 10 ft/s. In this case, steps 2 through 6 of the IRLS algorithm above are changed only by the replacement of v_i with v_0, one of the velocity values where predictions are desired.

Appendix B. Program Documentation & Comments

Below is the SAS code, preceded by a few notes, to fit binary data via the nonparametric local logistic regression procedure (llogr) presented in this report. The llogr procedure uses PRESS* as the method for bandwidth selection.

Some notes about the program are presented below.

1. Lines 6, 7, and 10 read the data from a file. The file can be a text (.txt), print (.prn), or flat data file. The program is set up to read a file containing two columns of data: the first column contains the velocity, and the second column is the response (0 or 1).

   The INFILE statement reads the data set from its stored location, denoted by "fi lename.filetype."

2. Once the data have been read into the IML procedure, the regressor (in this case, the velocity) is rescaled such that the values are between 0 and 1. The program then finds the bandwidth to minimize PRESS*. Once the bandwidth has been found, the program does the following:

   - Obtains the llogr fit at each of the data points (lines 220- 248)
   - Finds the lower and upper V_{50} values (lines 252- 452)
   - Obtains predictions at 10-ft/s intervals (lines 461- 502)

3. The program outputs the following (lines 518- 521):

   - The bandwidth (BW).
   - The chi-square statistic, the model degrees of freedom, and the mean squared error, denoted by CHISQR, MODELDF, and MSE, respectively.
   - The lower V_{50} (V_{50L}) and the upper V_{50} (V_{50U}) values.
   - The predictions at 10-ft/s intervals: Velocity (VEL), fitted value (YHAT0), and lower and upper confidence limits for the fitted value (LCL0 and UCL0).

4. Once the output is obtained, the user can produce a plot of the fitted curve using Excel or another plotting application. However, the SAS code below also generates a plot with upper and lower confidence bounds in Computer Graphic Metafile (CGM) format. The code is on lines 538 to 552. The user must supply the filename and the directory. DO NOT CHANGE THE EXTENSION. Once the CGM file has been created, the user can view the plot in MS Word or another application that converts CGM files.

SAS Code

The following represents the directory, filename, and filetype of the data set. The data set should be in text form or a flat data file.

```
Options ls=80 ps=54 nodate nonumber;

Title1 'Local Logistic Regression Analysis';
Title2 'using Press* to Obtain Bandwidth';

DATA MACRO;
   INFILE "d:\research\ARMY\MACROCOMPOSITE ORIGINAL DATA.PRN";
```

Here variables are being read from the data set. In this case, they are velocity and response.

```
   INPUT VELOCITY RESPONSE;

RUN;

DATA LLOGR;
   SET MACRO;
RUN;
PROC SORT;
BY VELOCITY;
RUN;
```

The following sorts the data set by velocity.

```
PROC IML;
   Use LLOGR;
   Read All Into X   Var {VELOCITY};
   Read All Into PI  Var {RESPONSE};
```

Now the velocity is being converted/transformed to the 0 to 1 data range. This is so that we can obtain a better interpretation of the bandwidth. That is, given a finite range for the bandwidth, one can interpret its relative size. However, if the data are not converted, the range of the bandwidth is from 0 to infinity, allowing a less meaningful interpretation.

```
N=NROW(X);
   MINX=MIN(X);
   MAXX=MAX(X);
   LENGTH=(MAXX-MINX)/10 + 1;
   VEL=J(LENGTH,1,0);
   V=0;
   DO I=MINX TO MAXX BY 10;
    V=V+1;
    VEL[V,]=(I-MINX)/(MAXX-MINX);
   END;
```

```
X=(X-MINX)/(MAXX-MINX); * Transforms X-variable to [0,1] Range;

*** INITIALIZATIONS ***;

H=J(N,1,1);
YHAT=J(N,1,1);
LCL=J(N,1,1);
UCL=J(N,1,1);
PRSS=J(3,1,0);
H0=J(N,1,0);
TYHATMI=J(N,1,1);
*** END OF INITIALIZATIONS ***;
```

Here begins the subroutine to obtain the true PRESS statistic and the subsequent bandwidth found to minimize this PRESS statistic.

```
START TRUEYMI;
     DO I=1 TO N;
      FREE XMI;
      IF I=1 THEN
        DO;
         XMI=X[2:N,];
         PMI=PI[2:N,];
        END;
      ELSE IF I=N THEN
          DO;
           XMI=X[1:N-1,];
           PMI=PI[1:N-1,];
          END;
      ELSE
        DO;
         XMI1=X[1:I-1,];
         PMI1=PI[1:I-1,];
         XMI2=X[I+1:N,];
         PMI2=PI[I+1:N,];
         XMI=XMI1//XMI2;
         PMI=PMI1//PMI2;
        END;
        XMI=J(N-1,1,1)||XMI;
        X0=1||X[I,];

        D0=X[I,]-XMI[,2];
        DSQ0=D0##2;
        DINT0 = (-1)#DSQ0/(BW##2);
        EXPD0 = EXP(DINT0);
```

```
        ESUMO = SUM(EXPDO); * ROW SUM OF NUMERATOR VALUES;
        KO = EXPDO/ESUMO;

        BHAT=INV(XMI`*DIAG(KO)*XMI)*XMI`*DIAG(KO)*PMI;
        DO ITER=1 TO 2;
         Y=XMI*BHAT;
         P=1/(1+EXP(-Y));
         Q=1-P;
         Z=EXP(-Y)/((1+EXP(-Y))##2);
         W=DIAG(P#Q)#DIAG(KO);
         XPXI=INV(XMI`*W*XMI);
         BHAT=BHAT+XPXI*XMI`*W*((PMI-P)/Z);
        END;
        TYHATMI[I,1]=1/(1+EXP(-XO*BHAT));
      END;
   FINISH;
* Subroutine to begin the search for the
  bandwidth that minimizes Press*;
START BWSEARCH;
      BWO = {0.1, 0.5, 0.7};
      DO A = 1 TO 3;
       BW=BWO[A,];
       D=X*J(1,N,1)-J(N,1,1)*X`;
       D=D`;
       DSQ=D##2;
       DINT = (-1)#(DSQ/(BW##2)#J(N,N,1));
       EXPD = EXP(DINT);
       ESUM = EXPD[,+]; * ROW SUM OF NUMERATOR VALUES;
       K = EXPD/(ESUM*J(1,N,1));

       RUN TRUEYMI;
       PRSS[A,] = SUM((PI-TYHATMI)##2)/(N-TRACE(K));
      END;

      * SECOND LOOP SEARCHES FOR THE BANDWIDTH TO MINIMIZE The True PRESS*;
      ITERB=0;
      MINI=MIN(PRSS);
      MINOLD=MAX(PRSS);
      DO WHILE (ABS(MINI-MINOLD)>1E-8);
       ITERB=ITERB+1;
       MINOLD=MAX(PRSS);
       IF MINI = PRSS[1,] THEN
         DO;
          AMIN = 1;
          SET1 = BWO[1,];
```

```
  SET2 = 2#BWO[1,]-BWO[2,];
  SET3 = BWO[1,]/2;
  VEC = SET3//SET2;
  BWO[1,] = MAX(VEC);
  BWO[3,] = BWO[2,];
  BWO[2,] = SET1;
 END;

 IF MINI = PRSS[2,] THEN
  DO;
   AMIN = 2;
   MAXI = MAX(PRSS);

   IF MAXI = PRSS[1,] THEN
     DO;
      AMAX = 1;
      BWO[1,] = BWO[2,];
      BWO[2,] = (BWO[2,] + BWO[3,])/2;
     END;

   IF MAXI = PRSS[3,] THEN
     DO;
      AMAX = 3;
      BWO[3,] = BWO[2,];
      BWO[2,] = (BWO[2,] + BWO[1,])/2;
     END;
  END;

 IF MINI = PRSS[3,] THEN
  DO;
   AMIN = 3;
   SET1 = BWO[3,];
   BWO[3,] = 2#BWO[3,]-BWO[2,];
   BWO[1,] = BWO[2,];
   BWO[2,]=SET1;
  END;

* OBTAIN THE LOCAL LOGISTIC FIT FOR THE NEW VALUE OF THE BANDWIDTH*;

BW = BWO[AMIN,];

D=X*J(1,N,1)-J(N,1,1)*X`;
D=D`;
DSQ=D##2;
DINT = (-1)#(DSQ/(BW##2)#J(N,N,1));
EXPD = EXP(DINT);
```

```
            ESUM = EXPD[,+]; * ROW SUM OF NUMERATOR VALUES;
            K = EXPD/(ESUM*J(1,N,1));

            RUN TRUEYMI;

            MINPRESS = SUM((PI-TYHATMI)##2)/(N-TRACE(K));
            * VALUE OF PRESS FOR THE CURRENT BANDWIDTH FOR THIS ITERATION*;

            * RESET POSITIONS OF BANDWIDTH AND PRSS VECTORS DEPENDING ON THE VALUE OF
             PRESS FOR THE CURRENT VALUE OF THE BANDWIDTH*;

            IF AMIN = 1 THEN
              DO;
               PRSS[3,] = PRSS[2,];
               PRSS[2,] = PRSS[1,];
               PRSS[1,] = MINPRESS;
              END;

            IF AMIN = 3 THEN
              DO;
               PRSS[1,] = PRSS[2,];
               PRSS[2,] = PRSS[3,];
               PRSS[3,] = MINPRESS;
              END;

            IF AMIN = 2 THEN
              DO;
               IF AMAX = 1 THEN PRSS[1,] = PRSS[2,];
               IF AMAX = 3 THEN PRSS[3,] = PRSS[2,];
               PRSS[2,] = MINPRESS;
              END;
            MINI=MIN(PRSS);
           END; * SEARCH FOR OPTIMAL BANDWIDTH ENDS;
        FINISH;

        WGT=1;
        RUN BWSEARCH;
     * Having obtained the bandwidth, we can now obtain the local logistic regression fit*;
     X2=J(N,1,1)||X;
        H=J(N,1,1);
        Do I=1 to N;
         D=X[I,]-X;
         DSQ=D##2;
         Dint=(-1)#DSQ/(BW##2);
         Expd=Exp(Dint);
         Esum=SUM(Expd);
```

```
     K=Expd/Esum;
     H[I,] = K[I,];

     BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
     DO ITER=1 TO 2;
       Y=X2*BHAT;
       P=1/(1+EXP(-Y));
       Q=1-P;
       Z=P#Q;
       W=DIAG(P#Q)#DIAG(K);
       XPXI=INV(X2`*W*X2);
       BHAT=bhat+XPXI*X2`*W*((PI-P)/Z);
     END;
     Y=X2*BHAT;
     P=1/(1+EXP(-Y));
     Q=1-P;
     Z=EXP(-Y)/((1+EXP(-Y))##2);
     W=DIAG(P#Q)#DIAG(K);
     XPXI=INV(X2`*W*X2);
     YHAT[I,]=1/(1+Exp(-X2[I,]*BHAT));
    END;
* ALGORITHM FOR COMPUTING THE LOWER V50 VALUE*;
COUNT=0;
    X50=J(3,1,0);
    DO I=1 TO N;
     IF COUNT=0 THEN
       DO;
        IF YHAT[I,]>0.5 THEN
         DO;
          X50[1,]=X[I,]-0.05;
          X50[2,]=X[I,]-0.025;
          X50[3,]=X[I,];
          COUNT=1;
         END;
       END;
    END;

    SET50 = J(3,1,0);

    * OBTAIN PHAT FOR 3 VALUES OF X (DOSE) *;

    DO I=1 TO 3;
     D=X50[I,]-X;
     DSQ=D##2;
     DINT=(-1)#DSQ/(BW##2);
```

```
EXPD=EXP(DINT);
ESUM=SUM(EXPD);
K=EXPD/ESUM;

BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
DO ITER=1 TO 2;
  Y=X2*BHAT;
  P=1/(1+EXP(-Y));
  Q=1-P;
  Z=EXP(-Y)/((1+EXP(-Y))##2);
  W=DIAG(P#Q)#DIAG(K);
  XPXI=INV(X2`*W*X2);
  BHAT=BHAT+XPXI*X2`*W*((PI-P)/Z);
 END;
 X0=1||X50[I,];
 SET50[I,]=1/(1+EXP(-X0*BHAT));
END;

PTEMP=0;
B=0;

DO B=1 TO 200;*WHILE (ABS(PTEMP-0.5)>1E-4);
 B=B+1;

 IF SET50[2,] > 0.5 THEN
  DO;
   INTERVAL=1;
   SET1 = X50[1,];
   SET2 = (X50[2,]+X50[1,])/2;
   SET3 = X50[2,];
   X50[1,] = SET1;
   X50[2,] = SET2;
   X50[3,] = SET3;
  END;

 IF SET50[2,] < 0.5 THEN
  DO;
   INTERVAL=2;
   SET1 = X50[2,];
   SET2 = (X50[3,] + X50[2,])/2;
   SET3 = X50[3,];
   X50[1,]=SET1;
   X50[2,]=SET2;
   X50[3,]=SET3;
  END;
```

```
* OBTAIN YHAT FOR NEW VELOCITY*;

V50L = X50[2,];

D=V50L-X;
DSQ=D##2;
Dint=(-1)#DSQ/(BW##2);
Expd=Exp(Dint);
Esum=SUM(Expd);
K=Expd/Esum;

BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
DO ITER=1 TO 2;
  Y=X2*BHAT;
  P=1/(1+EXP(-Y));
  Q=1-P;
  Z=EXP(-Y)/((1+EXP(-Y))##2);
  W=DIAG(P#Q)#DIAG(K);
  XPXI=INV(X2`*W*X2);
  BHAT=BHAT+XPXI*X2`*W*((PI-P)/Z);
END;
X0=1||V50L;
PTEMP=1/(1+Exp(-X0*BHAT));

IF INTERVAL=1 THEN SET50=SET50[1,]//PTEMP//SET50[2,];
IF INTERVAL=2 THEN SET50=SET50[2,]//PTEMP//SET50[3,];
END; * SEARCH FOR V50L ENDS;
IF (ABS(PTEMP-0.5)>1E-2) THEN V50L=.;
* ALGORITHM FOR COMPUTING THE UPPER V50 VALUE *;
COUNT=0;
DO I=N TO 1 BY (-1);
 IF COUNT=0 THEN
   DO;
    IF YHAT[I,]<0.5 THEN
     DO;
      X50[1,]=X[I,];
      X50[2,]=X[I,]+0.025;
      X50[3,]=X[I,]+0.05;
      COUNT=1;
     END;
   END;
END;

SET50 = J(3,1,0);
```

```
* OBTAIN PHAT FOR 3 VALUES OF X (DOSE) *;

DO I=1 TO 3;
 D=X50[I,]-X;
 DSQ=D##2;
 DINT=(-1)#DSQ/(BW##2);
 EXPD=EXP(DINT);
 ESUM=SUM(EXPD);
 K=EXPD/ESUM;

 BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
 DO ITER=1 TO 2;
   Y=X2*BHAT;
   P=1/(1+EXP(-Y));
   Q=1-P;
   Z=EXP(-Y)/((1+EXP(-Y))##2);
   W=DIAG(P#Q)#DIAG(K);
   XPXI=INV(X2`*W*X2);
   BHAT=BHAT+XPXI*X2`*W*((PI-P)/Z);
 END;
 X0=1||X50[I,];
 SET50[I,]=1/(1+EXP(-X0*BHAT));
END;

PTEMP=0;
B=0;
DO B=1 TO 200; *WHILE (ABS(PTEMP-0.5)>1E-4);
 B=B+1;

 IF SET50[2,] > 0.5 THEN
   DO;
    INTERVAL=1;
    SET1 = X50[1,];
    SET2 = (X50[2,]+X50[1,])/2;
    SET3 = X50[2,];
    X50[1,] = SET1;
    X50[2,] = SET2;
    X50[3,] = SET3;
   END;

 IF SET50[2,] < 0.5 THEN
   DO;
    INTERVAL=2;
    SET1 = X50[2,];
    SET2 = (X50[3,] + X50[2,])/2;
```

```
    SET3 = X50[3,];
    X50[1,]=SET1;
    X50[2,]=SET2;
    X50[3,]=SET3;
   END;

* OBTAIN YHAT FOR NEW VELOCITY*;

V50U = X50[2,];

D=V50U-X;
DSQ=D##2;
Dint=(-1)#DSQ/(BW##2);
Expd=Exp(Dint);
Esum=SUM(Expd);
K=Expd/Esum;

BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
DO ITER=1 TO 2;
  Y=X2*BHAT;
  P=1/(1+EXP(-Y));
  Q=1-P;
  Z=EXP(-Y)/((1+EXP(-Y))##2);
  W=DIAG(P#Q)#DIAG(K);
  XPXI=INV(X2`*W*X2);
  BHAT=BHAT+XPXI*X2`*W*((PI-P)/Z);
END;
X0=1||V50U;
Y=X2*BHAT;
P=1/(1+EXP(-Y));
Q=1-P;
Z=EXP(-Y)/((1+EXP(-Y))##2);
W=DIAG(P#Q)#DIAG(K);
XPXI=INV(X2`*W*X2);
PTEMP=1/(1+Exp(-X0*BHAT));

 IF INTERVAL=1 THEN SET50=SET50[1,]//PTEMP//SET50[2,];
 IF INTERVAL=2 THEN SET50=SET50[2,]//PTEMP//SET50[3,];
END; * SEARCH FOR V50U ENDS;
IF (ABS(PTEMP-0.5)>1E-2) THEN V50U=.;

MODELDF=SUM(H);
DFE=N-MODELDF;
CHISQR=SUM(((PI-YHAT)##2)/(YHAT#(1-YHAT)));
MSE=CHISQR/DFE;
```

```
*** MAKE PREDICTIONS AT INCREMENTS OF 10 FROM MIN VELOCITY TO MAX VELOCITY ***;
YHAT0=J(LENGTH,1,0);
  LCL0=J(LENGTH,1,0);
  UCL0=J(LENGTH,1,0);
  DO I=1 TO LENGTH;
   D=VEL[I,]-X;
   D=D`;
   DSQ=D##2;
   DINT = (-1)#DSQ/(BW##2);
   EXPD = EXP(DINT);
   ESUM = SUM(EXPD); * ROW SUM OF NUMERATOR VALUES;
   K = EXPD/ESUM;

   BHAT=INV(X2`*DIAG(K)*X2)*X2`*DIAG(K)*PI;
   DO iter = 1 to 2;
     Y=X2*BHAT;
     P=1/(1+EXP(-Y));
     Q=1-P;
     Z=P#Q;
     W=DIAG(P#Q)#DIAG(K);
     XPXI=INV(X2`*W*X2);
     BHAT=BHAT+XPXI*X2`*W*((PI-P)/Z);
   END;
   Y=X2*BHAT;
   P=1/(1+EXP(-Y));
   Q=1-P;
   Z=EXP(-Y)/((1+EXP(-Y))##2);
   W=DIAG(P#Q)#DIAG(K);
   XPXI=INV(X2`*W*X2);

   X0=1||VEL[I,];
   Y0=X0*BHAT;
   YHAT0[I,]=1/(1+EXP(-Y0));
   P0=YHAT0[I,];
   V0=DIAG(K##2)#DIAG(P#Q);
   VBHAT0=XPXI*X2`*V0*X2*XPXI;
   Z0=EXP(-Y0)/((1+EXP(-Y0))##2);
   VYHAT0=(Z0##2)#X0*VBHAT0*X0`;
   LCL0[I,]=YHAT0[I,]-1.96*SQRT(VYHAT0);
   UCL0[I,]=YHAT0[I,]+1.96*SQRT(VYHAT0);
   IF LCL0[I,]<0 THEN LCL0[I,]=0;
   IF UCL0[I,]>1 THEN UCL0[I,]=1;
  END;
```

```
VEL=VEL*(MAXX-MINX)+MINX;

C2={"X0" "YHAT0" "LCL0" "UCL0"};
PRED2=VEL||YHAT0||LCL0||UCL0;
CREATE YHAT2 FROM PRED2[COLNAME=C2];
   APPEND FROM PRED2;

X=X*(MAXX-MINX)+MINX;
V50L=V50L*(MAXX-MINX)+MINX;
V50U=V50U*(MAXX-MINX)+MINX;
```

Outputs are the bandwidth (BW), Chi-square statistic, model degrees of freedom, mean-squared error, V_{50} values, and the data (VEL, YHAT0, LCL0, and UCL0) at 10-ft/s intervals.

```
Print BW;
Print CHISQR MODELDF MSE /;
Print V50L V50U /;
PRINT VEL YHAT0 LCL0 UCL0;
```

The following creates a SAS data set that can be then used in the Proc Glot procedure.

```
C1={"X" "PI"};
 PRED1=X||PI;
 CREATE YHAT1 FROM PRED1[COLNAME=C1];
     APPEND FROM PRED1;

 DATA PHAT3;
    MERGE YHAT1 YHAT2;
 PROC SORT;
    BY X;
```

Below is the code to create a CGM format graph. This format gives us the capability to insert the plot in a Word file or any other application that reads the CGM file type.

```
FILENAME GSASFILE 'filename.cgm';
  GOPTIONS RESET=GOPTIONS DEVICE=CGM GACCESS=GSASFILE
       GPROTOCOL=SASGPASC GSFLEN=80;
  SYMBOL1 VALUE=DOT;
  SYMBOL2 INTERPOL=JOIN LINE=1 VALUE=NONE;
  SYMBOL3 INTERPOL=JOIN VALUE=NONE LINE=4;
  SYMBOL4 INTERPOL=JOIN VALUE=NONE LINE=4;
  AXIS1 LABEL=("VELOCITY, fps");
  AXIS2 ORDER=(0 TO 1 BY 0.1)
     LABEL=(ROTATE=90 ANGLE=-90 "RESPONSE");
  PROC GPLOT DATA=PHAT3;
```

```
    PLOT PI*X YHAT0*X0 UCL0*X0 LCL0*X0 / OVERLAY HAXIS=AXIS1 VAXIS=AXIS2;
      TITLE1 J=C H=2 'Local Logistic Regression Analysis';
      TITLE2 J=C H=2 'All Macrocomposite Data';
    RUN;
Quit;
```

Distribution

Admnstr
Defns Techl Info Ctr
ATTN DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

DARPA
ATTN S Welby
3701 N Fairfax Dr
Arlington VA 22203-1714

Ofc of the Secy of Defns
ATTN ODDRE (R&AT)
The Pentagon
Washington DC 20301-3080

Ofc of the Secy of Defns
ATTN OUSD(A&T)/ODDR&E(R) R J Trew
3080 Defense Pentagon
Washington DC 20301-7100

AMCOM MRDEC
ATTN AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

US Army TRADOC
Battle Lab Integration & Techl Dirctrt
ATTN ATCD-B
FT Monroe VA 23651-5850

Hdqtrs Dept of the Army
ATTN DAMO-FDT
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

US Military Acdmy
Mathematical Sci Ctr of Excellence
ATTN MADN-MATH MAJ M Huber
Thayer Hall
West Point NY 10996-1786

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
ATTN J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

SMC/CZA
2435 Vela Way Ste 1613
El Segundo CA 90245-5500

US Army ARDEC
ATTN AMSTA-AR-TD
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Info Sys Engrg Cmnd
ATTN AMSEL-IE-TD F Jenia
FT Huachuca AZ 85613-5300

US Army Natick RDEC Acting Techl Dir
ATTN SBCN-T P Brandler
Natick MA 01760-5002

US Army Simulation Train & Instrmntn Cmnd
ATTN AMSTI-CG M Macedonia
ATTN J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv Cmnd RDEC
ATTN AMSTA-TR J Chapin
Warren MI 48397-5000

Nav Surfc Warfare Ctr
ATTN Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

USAFAS
ATTN ATSF-CBL G Durham
FT Sill OK 73503-5600

Marine Corps Artillery Detachment
ATTN COL L Stuart
759 McNair Ave
Lawton OK 73503

Marine Corps Detachment
ATTN CPT Thomas
18B Sparrow Rd
Fallbrook CA 92028

Univ of Texas at Austin Inst for Advncd Tchnlgy
PO Box 202797
Austin TX 78720-2797

Distribution (cont'd)

British Aerospace Defence Sys Limited
ATTN P Wilkins
Vicarage Lane
Ilford Essex IG1 4Q
United Kingdom

Hicks & Assoc Inc
ATTN G Singley III
1710 Goodrich Dr Ste 1300
McLean VA 22102

Roke Manor Rsrch Roke Manor
ATTN I Sandoe
Romsey Hampshire S051 OZN
United Kingdom

US Army Rsrch Lab
ATTN AMSRL-CI-CD A Neiderer
ATTN AMSRL-CI-CD B Bodt (5 copies)
ATTN AMSRL-CI-CD C Hansen
ATTN AMSRL-CI-CD E Heilman
ATTN AMSRL-CI-CD G Moss
ATTN AMSRL-CI-CD J Forester
ATTN AMSRL-CI-CD J OMay
ATTN AMSRL-CI-CD M Thomas
ATTN AMSRL-CI-CD P Jones

US Army Rsrch Lab (cont'd)
ATTN AMSRL-CI-CD R C Kaste
ATTN AMSRL-CI-LP (305) (4 copies)
Aberdeen Proving Ground MD 21005

US Army Rsrch Lab
ATTN AMSRL-CS-IO-FI M Carr
Aberdeen Proving Ground MD 21005-5066

Director
US Army Rsrch Ofc
ATTN AMSRL-RO-D JCI Chang
PO Box 12211
Research Triangle Park NC 27709

US Army Rsrch Lab
ATTN AMSRL-CI-AI-R Mail & Records Mgmt
ATTN AMSRL-CI-AP Techl Pub (2 copies)
ATTN AMSRL-CI-LL Techl Lib (2 copies)
ATTN AMSRL-D D R Smith
ATTN AMSRL-DD J M Miller
ATTN AMSRL-SE-SA J Eicke
ATTN AMSRL-SE-SA R Frankel
ATTN AMSRL-SE-SA S Tenney
Adelphi Md 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE January 2001

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE Modeling Nonmonotonic Dose-Response Curves

5. FUNDING NUMBERS
DA PR: DY10
PE: 622783Y10

6. AUTHOR(S) Quinton J. Nottingham, Jeffrey B. Birch (Virgina Polytechnic Institute & State University), Barry A. Bodt (ARL)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-CI-CD email: babodt@arl.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1972

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 1TEDKC
AMS code: 622783Y1011

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A number of procedures have been used to analyze nonmonotonic binary data to predict the probability of response. Some classical procedures are the Up and Down strategy, the Robbins-Monro procedure, and other sequential optimization designs. Recently, nonparametric procedures such as kernel regression and local linear regression have been applied to this type of data. It is well known that kernel regression has problems fitting the data near the boundaries, and a drawback with local linear regression is that it may be "too linear" when fitting data from a curvilinear function. This report introduces a procedure called local logistic regression, which fits a logistic regression function at each of the data points. United States Army projectile data are used in an example that supports the use of local logistic regression for analyzing nonmonotonic binary data for certain response curves. Properties of local logistic regression are presented along with simulation results that indicate some of the strengths of the procedure.

14. SUBJECT TERMS Logistic regression local smoothing, nonparametric, nonmonotonic

15. NUMBER OF PAGES 40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102